

Introduction to Systems Programming

Systems software versus Application software

Classify the following: *wordprocessor, spreadsheet, video game, C compiler, Java compiler, Python interpreter, bash shell, Standard C library, GPU device driver, database software.*

- ▶ Operating systems are the quintessential systems programs.
- ▶ Systems programming often uses features specific to hardware devices.
- ▶ Systems programming often uses features and APIs specific to a given operating system.
- ▶ Systems programming deals with objects and concepts that are typically low-level.
- ▶ However concepts from systems programming are used in application programming and vice-versa.

Topics

- ▶ Simple systems programming.
- ▶ Using the file interface: POSIX, MS Windows Files.
- ▶ Creating and managing processes:
 - ▶ Creating multiple processes with Linux/UNIX system calls.
 - ▶ Creating multiple processes with MS Windows API calls.
- ▶ Creating and managing threads:
 - ▶ POSIX threads (Pthreads) multi-threading library.
 - ▶ Multi-threaded programming in Java
 - ▶ MS Windows API for threads

Using the Operating System

- ▶ A *file* is the fundamental unit of information storage.
- ▶ A *process* is the fundamental unit of computation.

A process uses *resources*. Examples of resources: CPU(s), GPU(s), memory, files, disk drives, tape drives, keyboards, display units, pipes, sockets etc.

The operating system maintains descriptors for processes and resources.

What is a Process?

- ▶ A program in execution.
- ▶ A process is a working structure, a (potentially) huge information refinery buzzing and blazing with activity as masses of information move around inside.
- ▶ A process is an information machine, merely enacted, temporarily embodied by an irrelevant hunk of metal, plastic and silicon called a computer. (From the book *Mirror Worlds...* by David Gelernter)
- ▶ A process is an abstraction (illusion?) maintained by the operating system.

What is a file?

- ▶ Files can be viewed as either:
 - ▶ a **sequence of bytes** with no structure imposed by the operating system.
 - ▶ or a **structured collection of information** with some structure imposed by the operating system.
- ▶ Unix, Linux, Windows, Mac OS X all treat files as a sequence of bytes. This model is the most flexible, leaving the structure of the file up to the application programs.

Code Examples

All of the code examples used in the class are available from the subversion repository via anonymous access with the following command:

```
svn co
```

```
https://cs.boisestate.edu/repos/csdept/courses/cs253/lab
```